NL030046

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/82301 A2

(51) International Patent Classification[7]: G11B 19/02, 7/00

(21) International Application Number: PCT/US01/12532

(22) International Filing Date: 16 April 2001 (16.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/552,288 19 April 2000 (19.04.2000) US

(71) Applicant: DATAPLAY, INC. [US/US]; 2560 55th Street, Boulder, CO 80301-5706 (US).

(72) Inventor: KEELER, Stanton, M.; 1156 Columbia Drive, Longmont, CO 80503 (US).

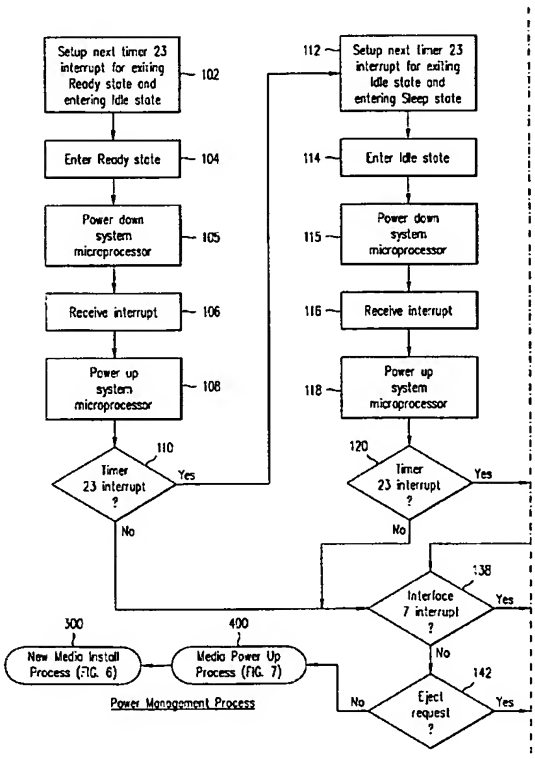
(74) Agents: STEUBER, David, E. et al.; Skjerven, Morrill, MacPherson, LLP, 25 Metro Drive, Suite 700, San Jose, CA 95110 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

*[Continued on next page]*

(54) Title: POWER MANAGEMENT FOR OPTICAL DRIVES




(57) Abstract: During a "Burst Read Mode", an optical storage device reads data from a medium into a buffer faster than a host device reads data out from the buffer. To improve power efficiency, the optical storage device powers down read/write components when a buffer fills to a first level. The read/write components include the optical pickup unit, the spindle motor, the actuator motor, their associated electronics, and the system microprocessor. While the read/write components are powered down, the host device continues to read data out of the buffer. When the output buffer empties to a second level, a buffer logic wakes up the system microprocessor and the system microprocessor powers up the read components. During a "Burst Write Mode", the host device reads data into the buffer slower than the optical storage device reads data out from the buffer to write the data to medium 3. To improves power efficiency, the optical storage device powers down read/write components when the buffer empties to a third level. While the read/write components are powered down, the host device continues to read data into the buffer. When the buffer fills to a fourth level, the buffer logic wakes up the system microprocessor and the system microprocessor powers up the write components. The optical storage device also transitions from one power state to another when a host device has not accessed the storage device within a specified time. In each transition, optical storage device powers down more and more read/write components until all the read/write components are powered down. Furthermore in the last power state, the buffer, which also stores the file system information from the medium, is also powered down.

WO 01/82301 A2

**Published:**
— *without international search report and to be republished upon receipt of that report*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

# POWER MANAGEMENT FOR OPTICAL DRIVES

## BACKGROUND

### Field of the Invention

This invention relates to a method and a system of power management for storage devices, and more particularly to power management of optical storage devices.

### Description of Related Art

Laser technology is integral to optical disk recording and storage systems. In such a system, mastering equipment can record digital data by burning a series of microscopic holes, commonly referred to as pits, with a laser beam into a thin metallic film on the surface of a small-diameter disc. In this manner, information is encoded on a master disc, which is replicated by a process known as stamping. Optical disk drives can also record digital data by modulating a laser light of high intensity to change the reflectivity of the disc surface of a writable optical disk.

To read the data, laser light of low intensity is reflected off the disc surface and is "read" by photodetectors. The amount of light received by the photodetectors varies according to the presence or the absence of the pits, or the change in the reflectivity of the disc surface, and this input is digitized by analog-to-digital circuits. The digital signals are subsequently converted back to analog information that can be displayed on a screen or played through speakers.

Power consumption plays a key part in the design of portable devices. Portable devices generally must be more efficient than non-portable devices because they run on batteries. Accordingly, power management plays a big role in the design of portable optical disk recording and storage systems.

## SUMMARY

During a "Burst Read Mode", a storage device reads data from a medium into a data buffer faster than a host device reads data out from the data buffer. The storage device improves power efficiency by powering down various read/write components used to read data from the medium into the data buffer, including the system

microprocessor, when the data buffer fills to a first level. While the various read/write components are powered down, the host device continues to read data out from the data buffer. When the data buffer empties to a second level, a buffer logic transmits an interrupt signal to wake the system microprocessor. Once awaken, the system microprocessor powers up the read/write components and reenters the Burst Read mode.

During a "Burst Write Mode", the host device reads data into the data buffer slower than the storage device reads data out from the data buffer to write the data onto the medium. The storage device improves power efficiency by powering down various read/write components when the data buffer empties to a third level. While the various read/write components are powered down, the host device continues to read data into the data buffer. When the data buffer fills to a fourth level, the buffer logic transmits the interrupt signal to wake the system microprocessor and reenter the Burst Write mode.

The storage device operates in various states of power consumption ("power state"). The storage device transitions from one power state to another power state if the storage device does not receive a command from the host device within specified times. In each of these transitions, the storage device powers down additional read/write components to save power. When the storage device detects an event, such as the receipt of a command from the host device or the insertion of a new medium, the optical storage device powers up the read/write components.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates in a block diagram a storage device in accordance with one aspect of the present invention.

FIG. 2 illustrates in a block diagram an embodiment of the storage device in FIG. 1.

FIG. 3 illustrates in a block diagram an implementation of the buffer logic in FIG. 2.

FIG. 4 illustrates in a flowchart an embodiment of the power management method.

FIG. 5 illustrates in a flowchart an embodiment of the Execute Command Process shown in FIG. 4.

FIG. 6 illustrates in a flowchart an embodiment of the New Media Install Process shown in FIG. 4 and FIG. 5.

FIG. 7 illustrates in a flowchart an embodiment of the Media Power Up Process shown in FIG. 4 and FIG. 5.

FIG. 8 illustrates in a flowchart an embodiment of the Burst Read Mode shown in FIG. 5.

FIG. 9 illustrates in a flowchart an embodiment of the Burst Write Mode shown in FIG. 5.

FIG. 10 illustrates in a flowchart an embodiment of the Laser Power Off Process shown in FIG. 5, FIG. 8, and FIG. 9.

FIG. 11 illustrates in a flowchart an embodiment of the Spin Motor Spindown Process shown in FIG. 5 and FIG. 8.

## DETAILED DESCRIPTION

FIG. 1 illustrates a storage device 1 in accordance with the invention. One embodiment of storage device 1 includes read/write device 2 that reads data from a medium 3 and writes the data to a data buffer 4. Read/write device 2 can also read data from data buffer 4 and write the data onto medium 3. Medium 3 is, for example, an optical disk storing compressed data such as Moving Picture Experts Group Audio Layer 3 (MP3). A host device 6 reads data out from data buffer 4 through a host interface 7. Host device 6 can also write data into data buffer 4 through host interface 7. Host interface 7 includes asynchronous and synchronous buses. Storage device 1 may be embedded in host device 6 or plugged in as an external peripheral device. Throughout the figures, double lines represent data paths, solid lines represent command paths, and dashed lines represent clock paths.

If host interface 7 is an asynchronous bus, host device 6 clocks host interface 7 with events, such as a read from or a write to medium 3, instead of a periodic clock signal. Such a host interface 7 may help to conserve power because host interface 7 is

not constantly being clocked and consuming power. Instead, host interface 7 is only clocked by host device 6 during certain events. Asynchronous bus design is understood by one skilled in the art and any conventional asynchronous bus design may be used to implement host interface 7. An asynchronous interface protocol between storage device 1 and host device 6 is described in a co-pending U.S. Patent Application Serial No. XX/XXX,XXX, attorney docket number M-8375 US, entitled "Asynchronous Input/Output Interface Protocol", assigned to the same assignee, which is hereby incorporated by reference.

In a Burst Read Mode of storage device 1, read/write device 2 reads data from medium 3 into data buffer 4 faster than host device 6 reads data out from data buffer 4. This may occur when the data read from medium 3 is compressed data such as MP3. Host device 6 reads data out from data buffer 4 at a slower rate because the compressed data contains a large quantity of data. One embodiment of storage device 1 powers down various components of read/write device 2 to conserve power (e.g., pauses its read of medium 3) when data buffer 4 fills to a first level (e.g., a first amount of data). While the various components of read/write device 2 are powered down, host device 6 continues to read data out from data buffer 4. Once data buffer 4 empties to a second level (e.g., a second amount of data), storage device 1 powers up the various components of read/write device 2 to continue to read data from medium 3 into data buffer 4.

In a Burst Write Mode, host device 6 reads data into data buffer 4 slower than storage device 1 reads data out from data buffer 4 to write the data onto medium 3. This may occur, for example, when host device 6 inputs digital audio data of low sampling rate into data buffer 4 for recording onto medium 3. Storage device 1 powers down various components of read/write device 2 when data buffer 4 empties to a third level (e.g., a third amount of data). While the various components of read/write device 2 are powered down, host device 6 continues to read data into data buffer 4. When data buffer 4 fills to a fourth level (e.g., a fourth amount of data), storage device 1 powers up the various components of read/write device 2 to continue to write data onto medium 3. Reference to "first", "second", "third", and "fourth" levels does not mean that all levels are different. For example, the "first" and "fourth" levels or the "second" and "third" levels could be the same in some embodiments.

In various power states, storage device 1 waits for commands from host device 6. If host device 6 does not issue a command (e.g., to read from or write to medium 3) within specified times, storage device 1 transitions from one power state to another power state to power down additional components of read/write device 2. In a transition to power state with the lost power consumption, data buffer 4 is also powered down. When storage device 1 detects an event, such as the receipt of a command from host device 6 or the insertion of a new medium 3, storage device 1 powers up the various components of read/write device 2.

Referring to FIG. 2, one embodiment of read/write device 2 includes a spindle motor 8 that spins medium 3, an optical pickup unit (OPU) 9 that reads data from medium 3, and an actuator motor 10 that moves OPU 9 radially along the tracks of medium 3. Spindle driver 11 controls the rotational speed of spindle motor 8. System microprocessor 12 controls spindle driver 11. To power down spindle motor 8, system microprocessor 12 drives a control signal Spindle Power Down to spindle driver 11 via an input/output line of system microprocessor 12. In response, spindle driver 11 stops the flow of current into spindle motor 8.

Another embodiment of spindle driver 11 does not stop the flow of current into spindle motor 8. Instead, this embodiment of spindle driver 11 causes spindle motor 8 to spin at a reduced speed. The reduced speed is, for example, 1000 RPM, as compared with the normal read/write speed of 2000 to 5000 RPM. Mechanical vibrations caused by stopping and starting the rotation of medium 3 are thereby avoided. In this embodiment, spindle driver 11 can operate spindle motor 8 at the idle speed even when clock 22 and system microprocessor 12 are powered down (described later). Spindle motor driver 11 is, for example, an ST 6254 spindle motor driver driving a Y wound spindle motor 8, which is available from STMicroelectronics NV of Saint Genis Pouilly, France.

Another embodiment of spindle driver 11 uses the spindle back-EMF of spindle motor 8 to provide interrupt signals to wake system microprocessor 12 from its powered down mode. When awakened by these interrupt signals, system microprocessor 12 issues commands to spindle driver 11 to maintain spindle motor 8 at the reduced speed.

Laser driver 13 controls a laser diode within OPU 9. To power down the laser of OPU 9, system microprocessor 12 drives a control signal OPU Power Down to laser driver 13 via an input/output line of system microprocessor 12. In response, laser driver 13 decouples OPU 9 from its power source. In one implementation, laser driver 13 uses a field effect transistor (FET) to couple or decouple OPU 9 from its power source.

OPU 9 passes data read from medium 3 to front-end electronics 14. Front-end electronics 14 filters the data and passes the filtered data to a data formatter 15. Data formatter 15 formats the data to a form that can be stored in data buffer 4 and output to host device 6 via host interface 7. Data buffer 4 can comprise one or a combination of several types of memory devices including DRAMs or SRAMs.

Actuator driver 16 controls the position of actuator motor 10. Servo microprocessor 17 controls actuator driver 16. Servo microprocessor 17 receives tracking information from front-end electronics 14. System microprocessor 12 controls servo microprocessor 17.

To power down actuator motor 10, system microprocessor 12 causes servo microprocessor 17 to drive a control signal Actuator Power Down to actuator driver 16 via an input/output line of servo microprocessor 17. In response to control signal Actuator Power Down, actuator driver 16 stops the flow of current into actuator motor 10.

Clock control 22 receives a master clock signal (e.g., 40 MHz) from crystal oscillator 33. Clock control 22 provides clock signals synchronized with the master clock signal to data buffer 4, system microprocessor 12, front-end electronics 14, data formatter 15, servo microprocessor 17, and timer 23. System microprocessor 12 can instruct clock control 22 to terminate clock signals to buffer 4, front-end electronics 14, data formatter 15, and servo microprocessor 17.

System microprocessor 12 can put itself in a powered down mode to conserve power consumption, i.e., the clock signal to system microprocessor 12 can be shut off. System microprocessor 12 awakes when it receives an interrupt at interrupt terminal 32. System microprocessor 12 can set timer 23 to send an interrupt ("timer interrupt") to interrupt terminal 32 to wake up system microprocessor 12 in a specified time.

Referring to FIG. 3, a buffer logic 5 can transmit an interrupt ("wake interrupt") to interrupt terminal 32. Buffer logic 5 includes a comparator 28 coupled to a counter 29 and a register 30. System microprocessor 12 can store a value in register 30 by outputting the value via a bus 41 coupled to a port 35 of register 30.

Counter 29 increments each time it receives at an input terminal 34 a read or write strobe signal from host device 6. Each time host device 6 strobes data buffer 4, a predetermined number of bytes is read into or out from data buffer 4. Thus, the number of times counter 29 increments corresponds to the number of bytes read into or out from data buffer 4 by host device 6. System microprocessor 12 has an input port 40 coupled to an output bus 37 of counter 34 to determine the value stored in counter 29. By determining the difference between the values of counter 29 at two points in time, system microprocessor 12 can determine the amount of data read into or out from data buffer 4 between the two points in time by host device 6.

Comparator 28 has an input port 36 coupled to output bus 37 (which carries the value stored in counter 34), and an input port 38 coupled to an output bus 39 of register 30 (which carries the value stored in register 30). When the value in counter 29 is equal to the value in register 30, comparator 28 sends the wake interrupt to interrupt terminal 32 to power up system microprocessor 12. Buffer logic 5 is used to measure the amount of data in data buffer 4 and to power up system microprocessor 12 when the amount of data in data buffer 4 fills or empties to certain levels.

Storage device 1 has four power states: Ready, Idle, Sleep, and Coma. Table 1 illustrates the components of read/write device 2 that are powered up or powered down in the four power states. When a component is powered down, that component consumes little or no power. Table 1 also shows the components of read/write device 2 that are clocked or not clocked in the four power states. When a component is not clocked, that component consumes little or no power.

Table 1

| | Ready | Idle | Sleep | Coma |
|---|---|---|---|---|
| Spindle Motor 8 | Powered Up | Powered Up | Powered Down | Powered Down |

| | | | | |
|---|---|---|---|---|
| Actuator Motor 10 | Powered Up | Powered Down | Powered Down | Powered Down |
| Laser of OPU 9 | Powered Up | Powered Down | Powered Down | Powered Down |
| Data Buffer 4 | Powered Up | Powered Up | Powered Up | Powered Down |
| Servo Microprocessor 17 | Clocked | Not Clocked | Not Clocked | Not Clocked |
| Front-end Electronics 14 | Clocked | Not Clocked | Not Clocked | Not Clocked |
| Data Formatter 15 | Not Clocked | Not Clocked | Not Clocked | Not Clocked |

The power state of storage device 1 progresses from Ready to Idle, from Idle to Sleep, and from Sleep to Coma in descending order of power consumption as additional components are powered down or not clocked. Storage device 1 transitions from a power state of higher power consumption to another power state of lower power consumption when storage device 1 does not detect certain events within a specified time. These events include the receipt of a command from host device 6, the receipt of a request to eject medium 3, and the receipt of a request to install a new medium 3.

Instead of immediately transitioning from the Ready State to the Coma State, storage device 1 may save power by transitioning through the Idle State where spindle motor 8 is still powered up and spinning medium 3. Idle State may help to conserve power because the power consumed to keep medium 3 spinning for a period of time may be less than the power consumed to stop spinning medium 3 and then start spinning medium 3 upon detecting one of the above-described events within that period of time. Continuously spinning medium 3 may also allow storage device 1 to react more quickly to a read or a write command from host device 6.

Storage device 1 may also save power by transitioning through the Sleep State where data buffer 4 continues to receive power. As described later, data buffer 4 stores

file system information read from medium 3 (e.g., file names, file locations, file sizes). If data buffer 4 is powered down, the file system information is lost. If host device 6 issues a command for storage device 1 to send file system information after data buffer 4 is powered down, storage device 1 will need to power up all the components to reread the file system information from medium 3 and to store the file system information into data buffer 4 (to "recover" the file system information). Accordingly, it may consume less power to keep data buffer 4 powered up for a period of time than to power down data buffer 4 and then recover the file system information from medium 3 upon receiving the send file system command from host device 6 within that period of time.

Referring to FIG. 4, the power management of storage device 1 (process 100) starts in action 102. In action 102, system microprocessor 12 sets a time value (e.g., 1 second) by which timer 23 sends the timer interrupt to interrupt terminal 32 of system microprocessor 12. This timer interrupt will bring system microprocessor 12 out of its powered down mode to transition storage device 1 from the Ready State to the Idle State to conserve power. In action 104, system microprocessor puts storage device 1 in the Ready State. In the Ready State, system microprocessor 12 leaves the laser of OPU 9, actuator motor 10, spindle motor 8, and data buffer 4 actively operating (e.g., powered up). System microprocessor 12 also instructs clock control 22 to provide clock signals to servo microprocessor 17 and front-end electronics 14, but not data formatter 15. To save power, data formatter 15 is only clocked when data is read from or written to medium 3. In the Ready mode, storage device 1 is ready to quickly move OPU 9 over a new location on medium 3 to read or write a file.

In action 105, system microprocessor 12 powers down. In action 106, system microprocessor 12 receives an interrupt at interrupt terminal 32. The interrupt can be the preset timer 23 interrupt or some other interrupt, e.g., an interrupt ("interface interrupt") received from host device 6 via host interface 7, an interrupt to request the ejection medium 3, or an interrupt to request the insertion of a new medium 3. In action 108, system microprocessor 12 powers itself up in response to the interrupt received in action 106. In action 110, system microprocessor 12 determines if the interrupt received is the timer 23 interrupt. If the interrupt received is the timer 23 interrupt, action 110 is followed by action 112. Otherwise, action 110 is followed by action 138.

In action 112, system microprocessor 12 sets a time value (e.g., 1 minute) by which timer 23 will send the timer interrupt to interrupt terminal 32 of system microprocessor. This timer interrupt powers up system microprocessor 12 to transition storage device 1 from the Idle State to the Sleep State. In action 114, system microprocessor 12 puts storage device 1 in the Idle State to conserve power. In the Idle State, system microprocessor 12 leaves the spindle motor 8 and data buffer 4 actively operating (e.g., powered up). Spindle motor 8 may spin medium 3 at a reduced spin speed as compare to the normal spin speed to conserve power. System microprocessor 12 also instructs clock control 22 to stop clocking servo microprocessor 17, front-end electronics 14, and data formatter 15.

In action 115, system microprocessor 12 powers down. In action 116, system microprocessor 12 receives an interrupt at interrupt terminal 32. In action 118, system microprocessor 12 powers up because of the interrupt received in action 116. In action 120, system microprocessor 12 determines if the interrupt received is the preset timer 23 interrupt. If the interrupt received is the timer 23 interrupt, action 120 is followed by action 122. Otherwise, action 120 is followed by action 138.

In action 122, system microprocessor 12 sets a time value (e.g., 30 minutes) by which timer 23 will send another timer interrupt to interrupt terminal 32 of system microprocessor 12. This timer interrupt will power system microprocessor 12 up to transition storage device 1 from the Sleep State to the Coma State. In action 124, system microprocessor 12 puts storage device 1 in the Sleep State to conserve more power. In the Sleep State, system microprocessor 12 leaves only data buffer 4 actively operating (e.g., powered up) to maintain any data stored therein, including file system information of medium 3. The file system information includes the names of the files, their locations (e.g., sector addresses) on medium 3, and their sizes. System microprocessor 12 also instructs clock control 22 to stop clocking servo microprocessor 17, front-end electronics 14, and data formatter 15. Th Sleep State is the first power state of storage device 1 where all the components of read/write device 2 are powered down.

In action 125, system microprocessor 12 powers itself down. In action 126, system microprocessor 12 receives an interrupt at interrupt terminal 32. In action 128, system microprocessor 12 powers up because of the interrupt received in action 126. In

action 130, system microprocessor 12 determines if the interrupt received is the preset timer 23 interrupt. If the interrupt received is the timer 23 interrupt, action 130 is followed by action 132. Otherwise, action 130 is followed by action 138.

In action 132, system microprocessor 12 puts storage device 1 in the Coma State to conserve the maximum power. In the Coma State, system microprocessor 12 powers down spindle motor, actuator motor, the laser of OPU 9, and data buffer 4. System microprocessor 12 also instructs clock control 22 to stop clocking servo microprocessor 17, front-end electronics 14, and data formatter 15. In the Coma State, system microprocessor 12 powers down data buffer 4 by disconnecting it from its power source. As a result, data buffer 4 loses any file system information of medium 3 stored therein. When host device 6 requests the file system information from medium 3, read/write device 2 must be powered up and the file system must be read again from medium 3. This recovery process is described later in reference to FIG. 6.

In action 133, system microprocessor 12 powers itself down. System microprocessor 12 continues in a powered down condition, and storage device 1 remains in the Coma State, until action 134, when system microprocessor 12 receives an interrupt at interrupt terminal 32. In action 136, system microprocessor 12 powers up because of the interrupt received in action 134.

In action 138, system microprocessor 12 determines if the interrupt received is an interrupt from host interface 7. An interface 7 interrupt indicates that host device 6 has issued or is about to issue a command, such as a read or a write command, for storage device 1 to execute. If the interrupt received is an interface 7 interrupt, action 138 is followed by an "Execute Command Process" (process 200) described later in reference to FIG. 5, in which system microprocessor 12 executes the command received from host device 6. Otherwise, action 138 is followed by action 142.

In action 142, system microprocessor 12 determines if the interrupt received is a request to eject medium 3 from host device 6 or a user. If the interrupt received is the eject request, action 142 is followed by action 144. Otherwise, action 142 is followed by a "Media Power Up Process" (process 400) described later in reference to FIG. 7, in which system microprocessor 12 powers up the components of storage device 1. Process 400 is followed by a "New Media Install Process" (process 300) described later in reference to FIG. 6, in which system microprocessor 12 processes the file system

information of new medium 3. In action 144, system microprocessor 12 instructs storage device 1 to physically eject medium 3. Action 144 is followed by the previously described action 132, where system microprocessor puts storage device 1 back in the Coma State.

As indicated in FIG. 4, the sequence beginning with action 138 may also be invoked if system microprocessor 12 receives a non-timer 23 interrupt when storage device 1 is in the Ready, Idle, or Sleep State.

FIG. 5 illustrates the Execute Command Process of storage device 1 (process 200). In action 202, system microprocessor 12 determines if the command from host device 6 is a command to send file system information ("send file system command") to host device 6. If the interface command is a send file system command, action 202 is followed by action 204. Otherwise, action 202 is followed by action 210.

In action 204, system microprocessor 12 determines if the file system information is available. The file system information is available if storage device 1 has saved the file system information previously read from medium 3 in data buffer 4. If the file system information is available, action 204 is followed by action 206. Otherwise, action 204 is followed by the Media Power Up Process 400 (FIG. 7). Process 400 is followed by the New Media Install Process 300 (FIG. 6). In action 206, system microprocessor 12 sends the file system information from data buffer 4 to host device 6 via host interface 7. Action 206 is followed by the previously described Power Management Process 100 (FIG. 4). In one embodiment, storage device 1 remains in the same power state it was at prior to entering Execute Command Process 200. Thus, FIG. 4 shows optional paths (dashed lines) that lead directly to the Idle State and the Sleep State.

In action 210, system microprocessor 12 determines if the command received is a command ("non-media command") that does not involve medium 3. Host device 6 uses non-media commands such as status commands to detect the presence of peripheral devices, configuration commands to set the data transmission size and the use of the power management algorithm described throughout this disclosure, and test commands to test data transmission to and from data buffer 4. If the command received is a non-media command, action 210 is followed by action 212. Otherwise, action 210 is followed by action 214.

In action 212, system microprocessor 12 processes the non-media command. Action 212 is followed by the previously described Power Management Process 100 (FIG. 4). In one embodiment, storage device 1 remains in the same power state it was at prior to entering process 200. Thus, action 212 is followed by one of actions 102, 112, and 122, depending on the prior power state of storage device 1.

In action 214, system microprocessor 12 determines if the command received is a read file command (e.g., to play an audio file). If the command received is not a read file command, the command received must be a write file command and action 214 is followed by action 216. If the command received is a read file command, action 212 is followed by the Media Power Up Process 400 (FIG. 7). Process 400 is followed the Burst Read Mode (process 500) described later in reference to FIG. 8, in which storage device 1 reads data from medium 3.

In action 216, system microprocessor 12 determines if OPU 9 is powered down. OPU is powered down if storage device 1 comes from a power state where the laser of OPU 9 is already powered down and OPU 9 is already parked. If OPU 9 is powered down, action 216 is followed by action 218. If OPU 9 is not powered down, action 216 is followed by a "Laser Power Off Process" (process 700) described later in reference to FIG. 10, in which system microprocessor 12 powers off the laser of OPU 9 and parks OPU 9 to conserve power. Process 700 is followed by action 218.

In action 218, system microprocessor 12 determines if motor 8 is spinning medium 3. If motor 8 is not spinning medium 3, action 218 is followed by the Burst Write Mode (process 600) described later in reference to FIG. 9, in which storage device 1 writes data to medium 3. Spindle motor 8 is not spinning if storage device 1 is in a power state where spindle motor 8 is spun down. If motor 8 is spinning medium 3, action 218 is followed by a "Spin Motor Spindown Process" (process 800) described later in reference to FIG. 11, in which system microprocessor 12 spins down spindle motor 8 to conserve power. Process 800 is followed by process 600.

FIG. 7 illustrates the Media Power Up Process 400. In action 402, system microprocessor 12 determines if spindle motor 8 is spinning. Spindle motor 8 is spinning if storage device 1 is in either of Ready State or Idle State. If spindle motor 8 is spinning, action 402 is followed by action 408. Otherwise, action 402 is followed by action 404. In action 404, system microprocessor 12 transmits signal Spindle Power

Down to spindle driver 11 via an input/output line of system microprocessor 12 ("enables" spindle driver 11) to cause spindle driver 11 to provide current to spindle motor 8. In action 406, spindle driver 11 provides current into spindle motor 8 and spins up spindle motor 8.

In action 408, system microprocessor 12 determines if storage device 1 is in the Ready State. If storage device 1 is in the Ready State, action 408 is followed by action 422. Otherwise, action 408 is followed by action 410. In action 410, system microprocessor 12 drives control signal OPU Power Down to laser driver 13 via an input/output line of system microprocessor 12. In response, laser driver 13 couples OPU 9 to its power source. In action 412, system microprocessor 12 causes clock control 22 to clock front-end electronics 14. In action 414, system microprocessor 12 causes clock control 22 to clock servo microprocessor 17.

In action 416, system microprocessor 12 causes servo microprocessor 17 to enable actuator driver 16. In response, servo microprocessor 17 drives control signal Actuator Power Down to actuator driver 16 via an input/output line of servo microprocessor 17. In response to control signal Actuator Power Down, actuator driver 16 provides current into actuator motor 10. In action 418, system microprocessor 12 causes OPU 9 to be un-parked (e.g., physically unlatched so OPU 9 can be moved by actuator motor 10 over the tracks of medium 3). In action 420, system microprocessor 12 changes the power state of storage device 1 to the Ready State. In action 422, storage device 1 exits the Media Power Up Process.

FIG. 6 illustrates the New Media Install Process 300. In action 302, system microprocessor 12 causes servo microprocessor 17 to place OPU 9 over the location of the file information on medium 3 to read the file system information. In action 304, system microprocessor 12 causes clock control 22 to clock data formatter 15. Also in action 304, OPU 9 passes the file system information to front-end electronics 14, and front-end electronics 14 passes the file system information to data formatter 15.

In action 306, system microprocessor 12 causes the file system information to be saved in data buffer 4 so host device 6 can read out the file system information. In action 308, system microprocessor 12 causes clock control 22 to stop clocking data formatter 15. Action 308 is followed by the previously described Power Management Process 100 (FIG. 4). File system information is further described in a co-pending U.S.

Patent Application Serial No. YY/YYY,YYY, attorney docket number M-8374 US, entitled "File System Management Embedded In A Storage Device", assigned to the same assignee, which is hereby incorporated by reference.

FIG. 8 illustrates the Burst Read Mode of storage device 1 (process 500). In action 502, system microprocessor 12 causes (1) clock control 22 to clock data formatter 15 and (2) read/write device 2 to read data from medium 3 into data buffer 4. In action 504, system microprocessor 12 determines if data buffer 4 has filled to the first level. As system microprocessor 12 controls the input of data from data formatter 15 into data buffer 4, system microprocessor 12 knows how much data was input in data buffer 4. If the input rate into buffer 4 is much greater than the output rate out of data buffer 4, system microprocessor 12 can assume that the amount of data in data buffer 4 at any given time is approximately the amount of data system microprocessor 12 has read into data buffer 4.

System microprocessor 12 can also determine the actual amount of data in data buffer 4 by subtracting the amount of data host device 6 has read out from data buffer 4 from the amount of data system microprocessor 12 has read into data buffer 4. System microprocessor 12 can determine the amount of data host device 6 has read out from data buffer 4 between any two given times by determining the difference between the values of counter 29 at the two given times. As previously described, counter 29 counts the amount of data read into or out of data buffer 4. Thus, system microprocessor 12 knows how much data host device 6 has read out from data buffer 4 between a time 0 (when system microprocessor 12 started reading data into data buffer 4) and a time 1 (the current time) by determining the difference between the values of counter 29 at time 0 and time 1. By subtracting the amount of data read out from data buffer 4 between time 0 and time 1 from the amount of data read into data buffer 4 between time 0 and time 1, system microprocessor 12 can determine the amount of data in data buffer 4 at time 1.

The first level is specified within the algorithm of system microprocessor 12. If buffer 4 has filled to the first level, action 504 is followed by action 506. Otherwise, action 504 is followed by action 518.

In action 506, system microprocessor 12 causes (1) read/write device 2 to stop reading data from medium 3 into data buffer 4 and (2) clock control 22 to stop clocking

data formatter 15 to conserve power and read/write device 2 to stop reading data from medium 3. In action 508, system microprocessor 12 reads counter 29 of buffer logic 5. In action 510, system microprocessor 12 writes a value to register 30. The value in register 30 is the sum of the value in counter 29 and a data buffer empty count. The data buffer empty count is the number of times counter 29 must be incremented for data buffer 4 to become empty. Thus, when the value of register 30 matches the value in counter 29 (e.g., when data buffer 4 empties to the second level), comparator 28 sends the wake interrupt to system microprocessor 12. Action 510 is followed by the Laser Power Off Process 700 (FIG. 10). Process 700 is followed by the Spin Motor Spindown Process 800 (FIG. 11). Process 800 is followed by action 512.

In action 512, system microprocessor 12 powers itself down to conserve power. Please note that after various components of read/write device 2 are powered down in process 700, process 800, and action 512 to conserve power, host device 6 continues to read data out via host interface 7 from data buffer 4, which remains powered on and clocked by clock control 22. In action 514, system microprocessor 12 receives an interrupt at interrupt terminal 32 and powers itself up. In action 516, system microprocessor 12 determines if the interrupt received is the wake interrupt from logic buffer 5. If the interrupt received is not a wake interrupt, action 516 is followed by the previously described Execute Command Process 200 (FIG. 5). If the interrupt received is a wake interrupt, action 516 is followed by the previously described Media Power Up Process 400 (FIG. 7). Process 400 is followed by action 502.

In action 518, system microprocessor 12 determines if the requested file has been completely read into data buffer 4. System microprocessor 12 knows the size of the file being read from the file system information retrieved from medium 3 in the New Media Install Process 300 (FIG. 6). The requested file has been completely read into data buffer 4 if the amount of data read from medium 3 is equal to the requested file size. If the requested file has been completely read, action 518 is followed by action 520. Otherwise, action 518 is followed by action 504 and the algorithm cycles until either data buffer 4 fills to the first level or the file is completely read. In action 520, system microprocessor 12 causes clock control 22 to stop clocking data formatter 15. Action 520 is followed by the previously described Power Management 100 (FIG. 4).

FIG. 9 illustrates the Burst Write Mode of storage device 1 (process 600). In action 602, system microprocessor 12 reads counter 29 in buffer logic 5 (FIG. 3). In action 604, system microprocessor 12 writes a value to register 30. The value written in register 30 is the sum of the value read from counter 29 and a data buffer fill count or a file end count. The data buffer fill count is the number of times counter 29 must be incremented for data buffer 4 to become full. Once data buffer 4 is full, buffer logic 5 must wake system microprocessor 12 to read data out from data buffer 4 to prevent an overflow of data in data buffer 4.

Generally, the size of the file is not equal to the size of data buffer 4 and the last portion of the file does not fill the entire data buffer 4. The file end count is the number of times counter 29 must be incremented for the last portion of the file to be completely transferred from host device 6 to data buffer 4. Once the file has been completely transferred to data buffer 4, buffer logic 5 will wake system microprocessor 12 to read data out from data buffer 4 and finish writing the file to medium 3.

Host device 6 provides the size of the file to be written to medium 3 in the beginning of the data transfer to data buffer 4 via host interface 7. System microprocessor 12 can read the file size from data buffer 4 and determine the number of times data buffer 4 must be filled before the file is completely transferred. System microprocessor 12 can also use the file size to determine the file end count. The following example illustrates the use of the buffer fill count and the file end count, where data buffer 4 stores "x" bytes of data, a file has a size of "8x + 3y" bytes (where x = 500,000y), and counter 29 increments each "y" bytes of data transferred from host device 6 to data buffer 4.

When system microprocessor 12 receives the file size, it knows that data buffer 4 must be filled at least 9 times ($9x > 8x + 3y$) before the file is completely transferred. Thus, in the first eight passes through action 604, system microprocessor 12 writes in register 30 the sum of the buffer fill count (i.e., 500,000) and the current count in counter 29. In the ninth pass through action 604, system microprocessor 12 knows that there are only "3y" bytes of the file that must be transferred from host device 6 to data buffer 4. Thus, system microprocessor 12 writes in register 30 the sum of the end file count (i.e., 3) and the current count in counter 29.

In summary, when the value in counter 29 matches the value of register 30 (e.g., when data buffer 4 fills to the fourth level), comparator 28 sends the wake interrupt to system microprocessor 12.

In action 606, system microprocessor 12 powers itself down. Please note that after various components of read/write device 2 are powered down in process 700, process 800, and action 606 to conserve power, host device 6 continues to read data via host interface 7 into data buffer 4, which remains powered up and clocked by clock control 22. In action 608, system microprocessor 12 receives an interrupt at interrupt terminal 32 and powers itself up. In action 610, system microprocessor 12 determines if the interrupt received is the wake interrupt from buffer logic 5, indicating that data buffer 4 is full or the file has been completely transferred to data buffer 4. If the interrupt received is the wake interrupt, action 610 is followed by the previously described Media Power Up Process 400. Otherwise, action 610 is followed by the previously described Execute Command Process 200. Process 400 is followed by action 612. In action 612, system microprocessor 12 causes (1) clock control 22 to clock data formatter 15 and (2) read/write device 2 to write data from data buffer 4 onto medium 3.

In action 614, system microprocessor 12 determines if data buffer 4 has emptied to the third level. As system microprocessor 12 controls the read of data out from data buffer 4 to data formatter 15, system microprocessor 12 knows how much data was read out from data buffer 4. System microprocessor 12 can also determine the amount data host device 6 has read in data buffer 4 by determining the difference between the values of counter 29 at any two given times 5. As previously described, counter 29 counts the amount of data read into data buffer 4. Thus, system microprocessor12 knows how much data host device 6 has read into data buffer 4 between a time 2 (when system microprocessor 12 started reading data out from data buffer 4) and a time 3 (the current time) by determining the difference between the values of counter 29 at time 2 and time 3. By subtracting the amount of data read out from data buffer 4 between time 2 and time 3 from the amount of data read into data buffer 4 between time 2 and time 3, system microprocessor 12 can determine the amount of data in data buffer 4 at time 3.

The third level is specified within the algorithm of system microprocessor 12. If the data buffer has emptied to the third level, action 614 is followed by action 616.

Otherwise, action 614 is followed by action 618. In action 616, system microprocessor 12 causes (1) read/write device 2 to stop writing data to medium 3 and (2) clock control 22 to stop clocking data formatter 15. Action 616 is followed by the Laser Power Off Process 700 (FIG. 10). Process 700 is followed by action 602.

In action 618, system microprocessor 12 determines if the requested file has been completely written to medium 3. As previously described, system microprocessor 12 knows the file size and thus can determine if it has written the entire file onto medium 3. If the requested file has been completely written, action 618 is followed by action 620. Otherwise, action 618 is followed by action 614 and the program cycles until either data buffer 4 fills to the third level or the file is completely written to medium 3. In action 620, system microprocessor 12 causes (1) read/write device 2 to stop writing data to medium 3 and (2) clock control 22 to stop clocking data formatter 15. Action 620 is followed by the previously described Power Management 100 (FIG. 4).

FIG. 10 illustrates the Laser Power Off Process (process 700) used to conserve power. In action 702, system microprocessor 12 causes actuator motor 10 to move OPU 9 to a park position where OPU 9 is physically restrained. In action 704, system microprocessor 12 causes actuator driver 16 to stop the current flow to actuator motor 10. In action 706, system microprocessor 12 causes clock control 22 to stop clocking servo microprocessor 17. In action 708, system microprocessor 12 causes clock control 22 to stop clocking font-end electronics 14. In action 710, system microprocessor 12 causes the laser driver 13 to remove the power from the laser of OPU 9. In action 712, system microprocessor 12 exits process 700.

FIG. 11 illustrates the Spin Motor Spindown Process (process 800). In action 802, system microprocessor 12 causes spindle driver 11 to spindown spin motor 8. In action 804, system microprocessor 12 causes spindle driver 11 to stop providing current to spindle motor 8. In action 806, system microprocessor 12 exits process 800.

Although the invention has been described with reference to particular embodiments, the description is only an example of the invention's application and should not be taken as a limitation. For example, although the disclosure discusses the use of an optical medium and an optical pickup unit, a magnetic medium and a magnetic pickup unit can also be used. In addition, although exemplary transitions

between power states are provided in the disclosure, host device 6 may command storage device 1 to skip various power states by issuing configuration commands. Furthermore, although various components of storage device 1 are shown connected through dedicated data or control lines, one skilled in the art understands that these lines may represent a bus used by the components of storage device 1 to communicate data and commands. Various other adaptations and combinations of features of the embodiments disclosed are within the scope of the invention as defined by the following claims.

CLAIMS

I CLAIM:

1. A method to manage power consumption of a storage device, the method comprising:

reading data from a medium with a read device;

storing the data in a buffer;

outputting the data from the buffer; and

turning off at least partially the read device when the buffer fills to a first level while continuing to output data from the buffer.

2. The method of Claim 1, wherein said turning off comprises decoupling a laser of an optical pickup unit from a power source.

3. The method of Claim 1, wherein said turning off comprises stopping a flow of current to a spindle motor.

4. The method of Claim 1, wherein said turning off comprises stopping a flow of current to an actuator motor.

5. The method of Claim 1, wherein said turning off comprises terminating the clocking of a data formatter.

6. The method of Claim 1, wherein said turning off comprises terminating the clocking of a front-end electronics.

7. The method of Claim 1, wherein said turning off comprises terminating the clocking of a servo processor.

8. The method of Claim 1, wherein said turning off comprises putting a system processor into a powered down mode.

9. The method of Claim 1, wherein said medium includes an optical disk.

10. The method of Claim 1, wherein said turning off comprises:

determining a first amount of data read into the buffer between a first time and a second time;

determining a second amount of data read out from the buffer by:

determining a difference between a first value recorded in a counter at the first time from a second value recorded in the counter at a second time, wherein said counter is coupled to receive read strobe signals used to read a predetermined number of data from the buffer; and

multiplying the difference by the predetermined number of data; and

subtracting the second amount from the first amount.

11. The method of Claim 1, further comprising:

turning on at least partially the read device when the buffer empties to a second level.

12. The method of Claim 1, wherein said turning on comprises:

storing a first value in a register, wherein a comparator is coupled to receive the first value stored in the register and a second value stored in a counter, and the counter is coupled to receive read strobe signals used to read data out from the buffer;

powering down a system microprocessor;

receiving a wake interrupt from the comparator; and

powering up the system microprocessor.

13. The method of Claim 12, wherein said the first value corresponds to a number of times the counter must increment for the buffer to empty to the second level.

14. The method of Claim 11, further comprising:

subsequent to said turning on, reading data from the medium with the read device.

15. The method of Claim 1, wherein said storing and said outputting occur concurrently.

16. The method of Claim 1, wherein said storing occurs prior to said outputting.

17. The method of Claim 1, further comprising:

prior to said reading data, reading file system information from the medium with the read device; and

subsequent to said reading file system information, transmitting the file system information to a host device.

18. The method of Claim 1, further comprising:

receiving a request for the data from a host device.

19. The method of Claim 1, wherein the data includes compressed data.

20. The method of Claim 19, wherein the compressed data includes compressed audio data.

21. A method to manage power consumption of a storage device, the method comprising:

receiving data in a buffer;

reading the data out from the buffer;

writing the data to a medium with a write device; and

turning off the write device when the buffer empties to a first level.

22. The method of Claim 21, wherein said turning off comprises decoupling a laser of an optical pickup unit from a power source.

23. The method of Claim 21, wherein said turning off comprises stopping a flow of current to a spindle motor.

24. The method of Claim 21, wherein said turning off comprises stopping a flow of current to an actuator motor.

25. The method of Claim 21, wherein said turning off comprises terminating the clocking of a data formatter.

26. The method of Claim 21, wherein said turning off comprises terminating the clocking of a front-end electronics.

27. The method of Claim 21, wherein said turning off comprises terminating the clocking of a servo processor.

28. The method of Claim 21, wherein said turning off comprises putting a system processor into a powered down mode.

29. The method of Claim 21, further comprising receiving in the buffer file system information including the size of the data prior to said receiving data.

30. The method of Claim 21, wherein said turning off comprises:

storing a first value in a register, wherein a comparator is coupled to receive the first value stored in the register and a second value stored in a counter, and the counter is coupled to receive read strobe signals used to read data out from the buffer;

powering down a system microprocessor;

receiving a wake interrupt from the comparator; and

powering up the system microprocessor.

31. The method of Claim 30, wherein the first value corresponds to a number of times the counter must increment for the buffer to fill to a second level.

32. The method of Claim 30, wherein the first value corresponds to a number of times the counter must increment for the data to be completely transferred to the buffer.

33. The method of Claim 21, wherein said medium is an optical disk.

34. The method of Claim 21, further comprising turning on the read device when the buffer fills to a second level.

35. The method of Claim 34, further comprising, subsequent to said turning on, writing data to the medium with the write device.

36. The method of Claim 21, wherein said storing and said writing occur concurrently.

37. The method of Claim 21, wherein said storing occurs prior to said writing.

38. A method of managing power for storage devices, the method comprising:

reading file system information with a read device from a first medium;

storing the file system information in a buffer;

transmitting file system information from the buffer to a host device; and

turning off at least partially the read device if the host device does not transmit a command to the read device within a predetermined time.

39. The method of Claim 38, wherein the file system information includes a file name, a sector address, and a file size.

40. The method of Claim 38, wherein said turning off comprises decoupling a laser of an optical pickup unit from a power source.

41. The method of Claim 38, wherein said turning off comprises stopping a flow of current to a spindle motor.

42. The method of Claim 38, wherein said turning off comprises stopping a flow of current to an actuator motor.

43. The method of Claim 38, wherein said turning off comprises terminating the clocking of a data formatter.

44. The method of Claim 38, wherein said turning off comprises terminating the clocking of a front-end electronics.

45. The method of Claim 38, wherein said turning off comprises terminating the clocking of a servo processor.

46. The method of Claim 38, wherein said turning off comprises putting a system processor into a powered down mode.

47. The method of Claim 38, wherein said turning off comprises decoupling the buffer from a power source.

48. The method of Claim 38, wherein the first medium is an optical disk.

49. The method of Claim 38, further comprising receiving a command from the host device and turning on the read device.

50. The method of Claim 38, further comprising receiving a signal indicating an event and turning on the read device.

51. The method of Claim 50, further comprising reading file system information from the first medium.

52. The method of Claim 50, wherein the event is the insertion of a second medium.

53. The method of Claim 52, further comprising reading file system information from the second medium.

54. The method of Claim 38, further comprising:

subsequent to said transmitting file system information, setting a timer to send a timer interrupt after a specified time;

powering down a system microprocessor subsequent to said turning off;

receiving a timer interrupt; and

powering up the system microprocessor.

55. An optical storage drive comprising:

a system microprocessor having an interrupt terminal;

a buffer; and

a buffer logic comprising:

a counter having:

a first input terminal coupled a read or write strobe signal to the buffer; and

a first output bus;

a register having:

a second output bus; and

a comparator having:

a first input port coupled to the first output bus;

a second input port coupled to the second output bus; and

a first output line coupled to the interrupt terminal.

56. The optical storage drive of Claim 55, further comprising a timer having a third input port coupled to the system microprocessor.

57. The optical storage drive of Claim 56, wherein the timer has a second output line coupled to the interrupt terminal.

58. The optical storage drive of Claim 55, further comprising a timer having a second output line coupled to the interrupt terminal.

59. The optical storage drive of Claim 55, wherein the interrupt terminal is further coupled to receive an interrupt signal from a host device.

60. The optical storage device of Claim 55, further comprising an asynchronous bus coupling the buffer to a host device.

61. An optical storage drive comprising:

a read/write device;

a host interface;

a buffer having stored therein:

a size of a file on a medium;

a location of the file;

a name of the file.

62. The optical storage device of Claim 61, wherein said read/write device includes a system microprocessor.

63. The optical storage device of Claim 62, further comprising:

a buffer logic comprising:

a counter having:

a first input terminal coupled a read or write strobe signal to the buffer; and

a first output bus;

a register having:

a second output bus; and

a comparator having:

a first input port coupled to the first output bus;

a second input port coupled to the second output bus; and

a first output line coupled to the interrupt terminal.

64. The optical drive of Claim 61, wherein the host interface is an asynchronous bus.

65. The optical storage drive of Claim 62, wherein the read/write device further includes a timer having an input port coupled to the system microprocessor.

66. The optical storage drive of Claim 65, wherein the system microprocessor includes an interrupt terminal, the timer further having an output line coupled to the interrupt terminal, and the second output line carries a timer interrupt signal.

67. The optical storage drive of Claim 62, wherein the system microprocessor includes an interrupt terminal, the read/write device further includes a timer having an output line coupled to the interrupt terminal, the output line carrying a timer interrupt signal.

68. The optical storage drive of Claim 61, wherein the interrupt terminal is further coupled to receive an interrupt signal from a host device.

69. The optical storage device of Claim 61, further comprising an asynchronous bus coupling the buffer to a host device.

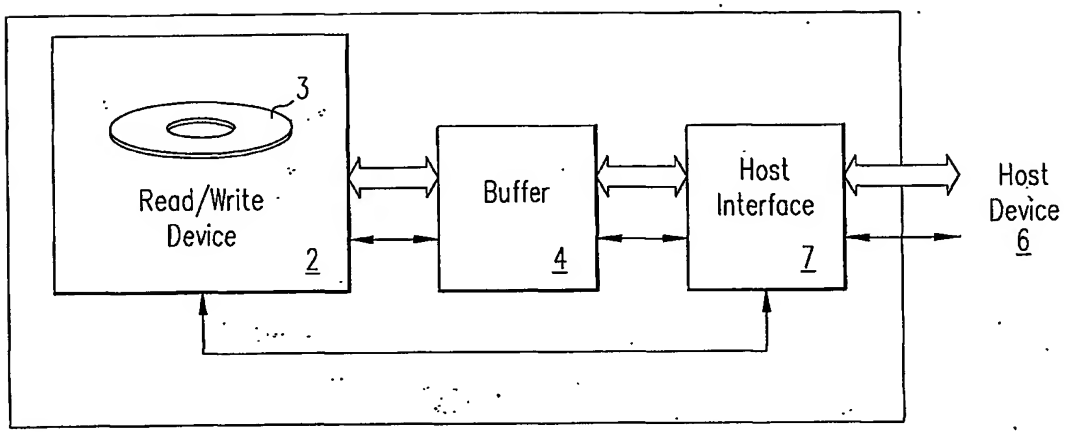
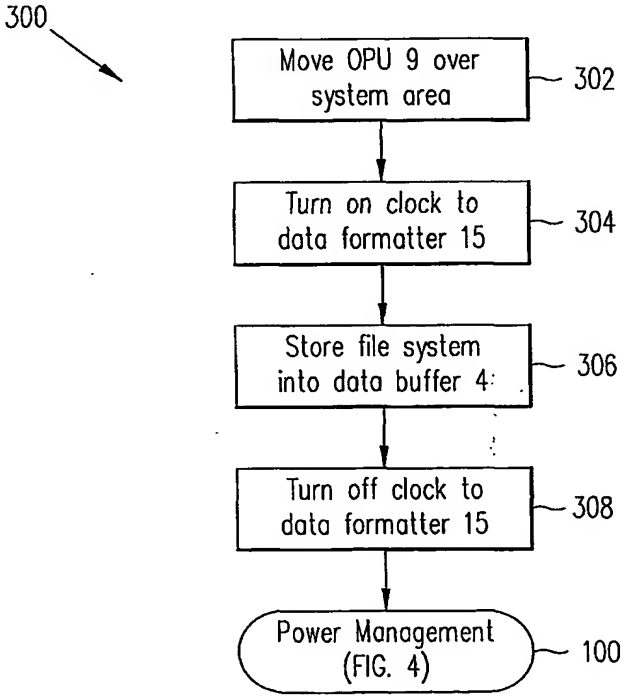



FIG. 1




New Media Install Process

FIG. 6

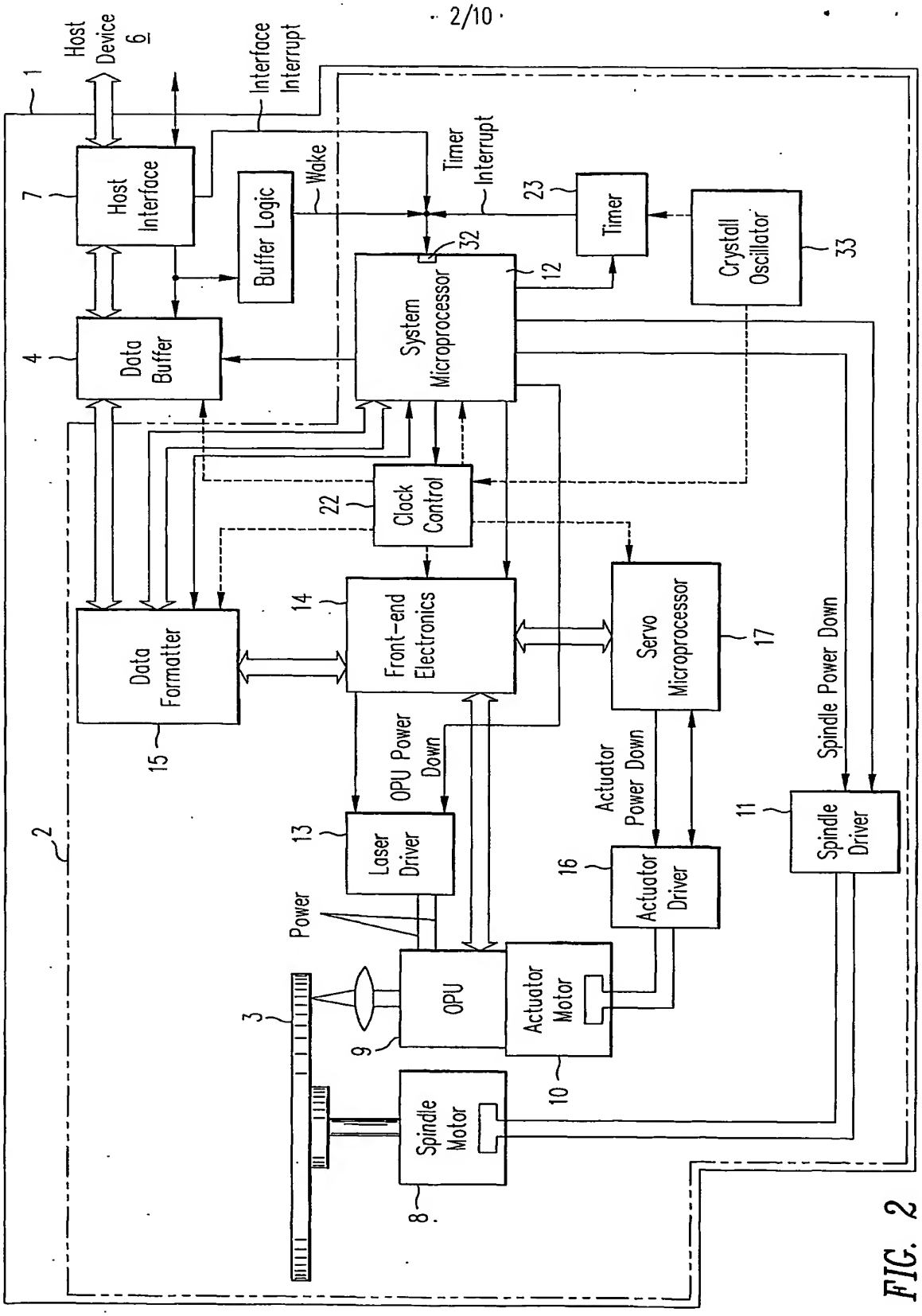

Host Device 6
Interface Interrupt
1
Host Interface
7
Buffer Logic
Wake
Timer Interrupt
23
Timer
Crystall Oscillator
33
32
12
System Microprocessor
Data Buffer
4
22
Clock Control
14
Front-end Electronics
Servo Microprocessor
17
Data Formatter
15
OPU Power Down
Actuator Power Down
Spindle Power Down
11
Spindle Driver
2
13
Laser Driver
16
Actuator Driver
Power
3
OPU
9
Actuator Motor
10
Spindle Motor
8

FIG. 2

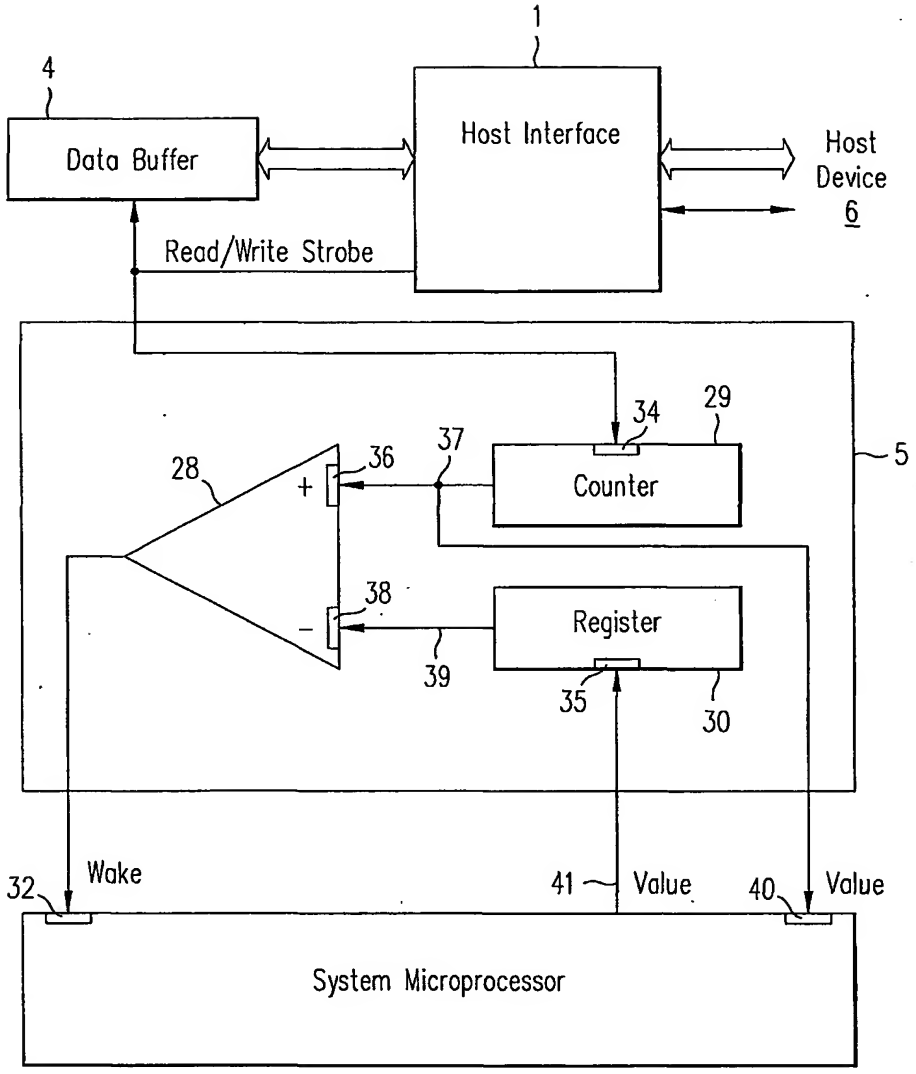

1
4
Data Buffer
Host Interface
Host Device 6
Read/Write Strobe
5
34
29
37
28
36
+
Counter
38
Register
39
35
30
32
Wake
41
Value
40
Value
System Microprocessor

*FIG. 3*

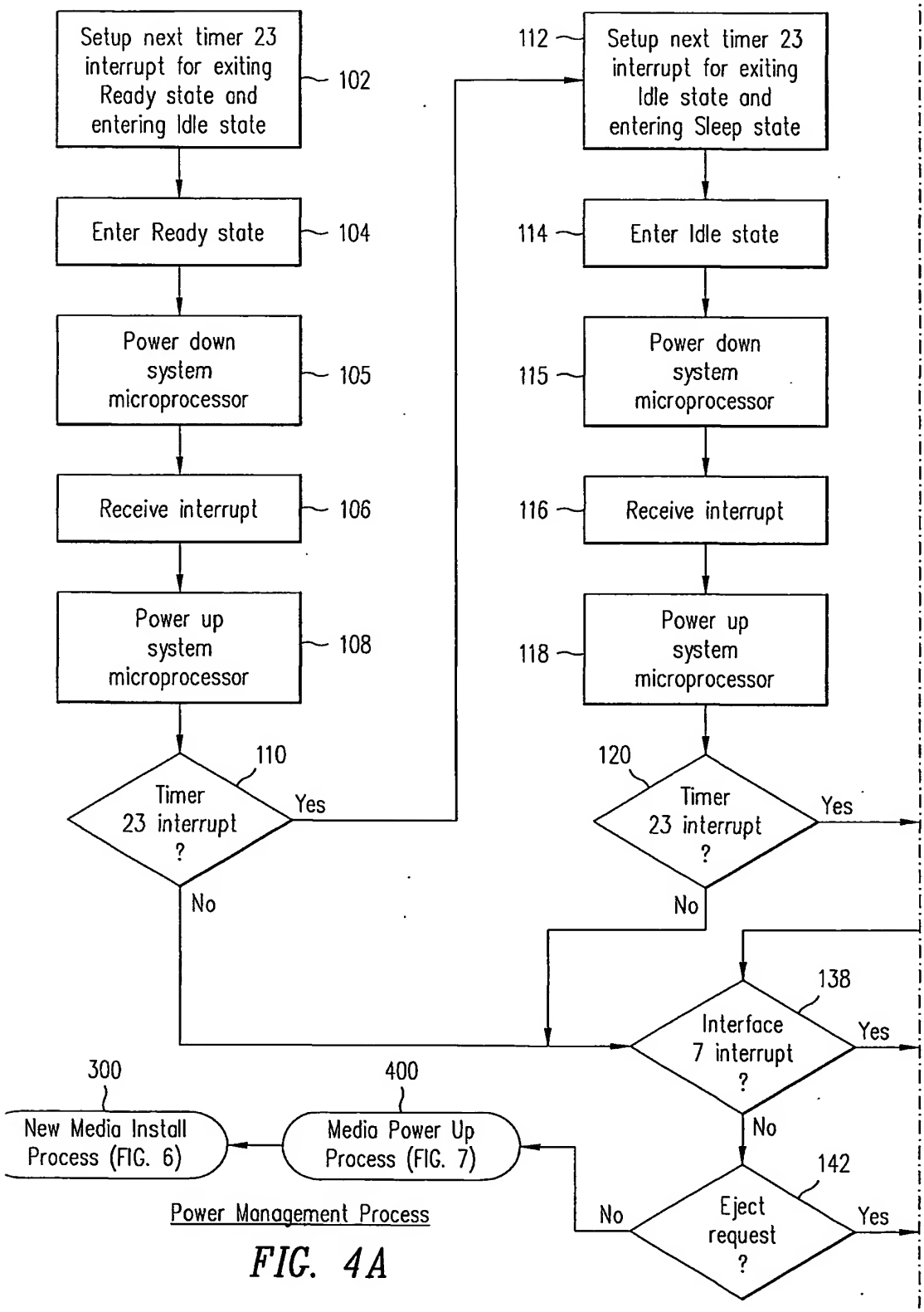

Setup next timer 23 interrupt for exiting Ready state and entering Idle state
102
Enter Ready state
104
Power down system microprocessor
105
Receive interrupt
106
Power up system microprocessor
108
110
Timer 23 interrupt ?
Yes
No
112
Setup next timer 23 interrupt for exiting Idle state and entering Sleep state
114
Enter Idle state
115
Power down system microprocessor
116
Receive interrupt
118
Power up system microprocessor
120
Timer 23 interrupt ?
Yes
No
138
Interface 7 interrupt ?
Yes
No
142
Eject request ?
Yes
No
300
New Media Install Process (FIG. 6)
400
Media Power Up Process (FIG. 7)
Power Management Process

*FIG. 4A*

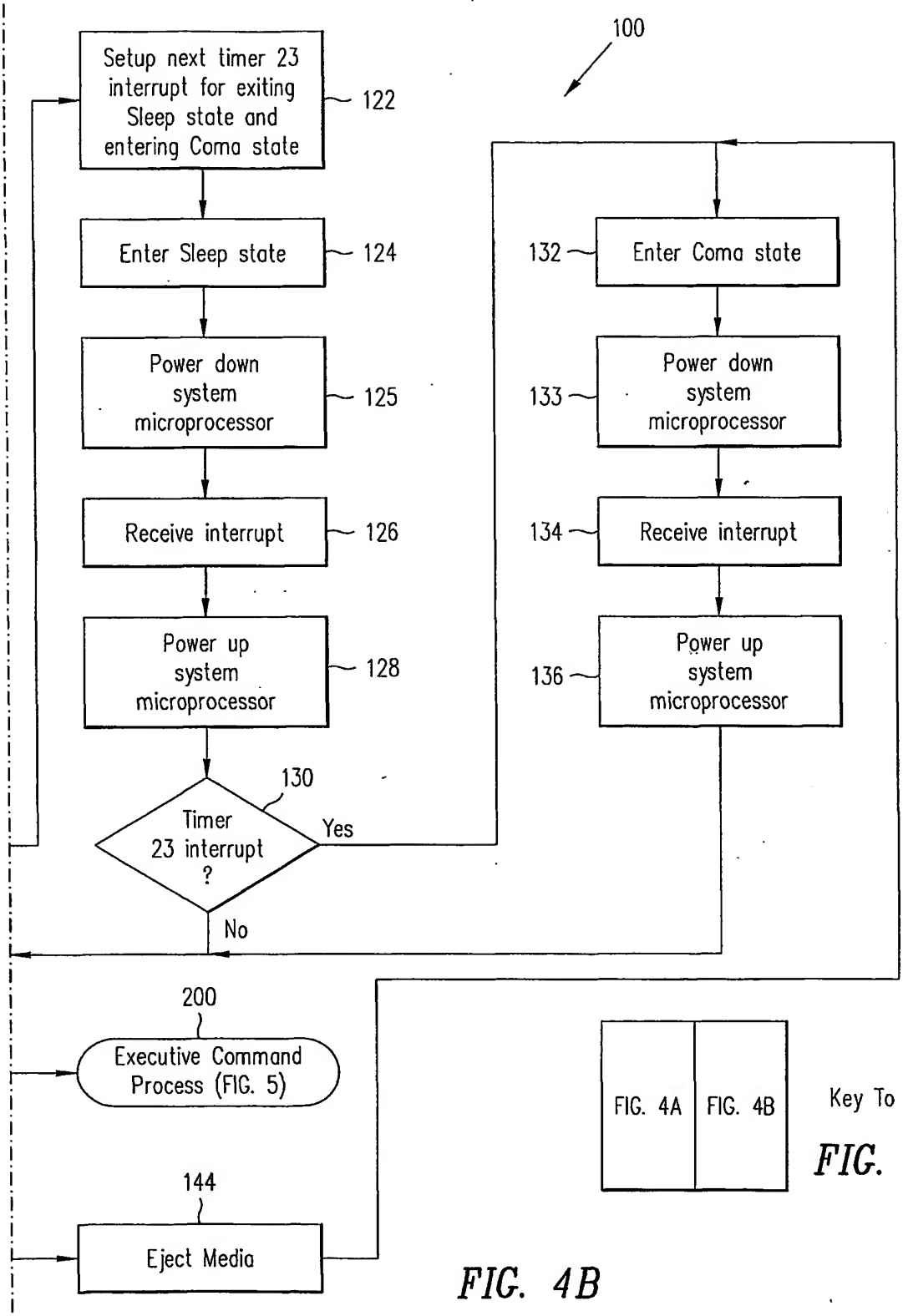



FIG. 4B




Key To

FIG. 4

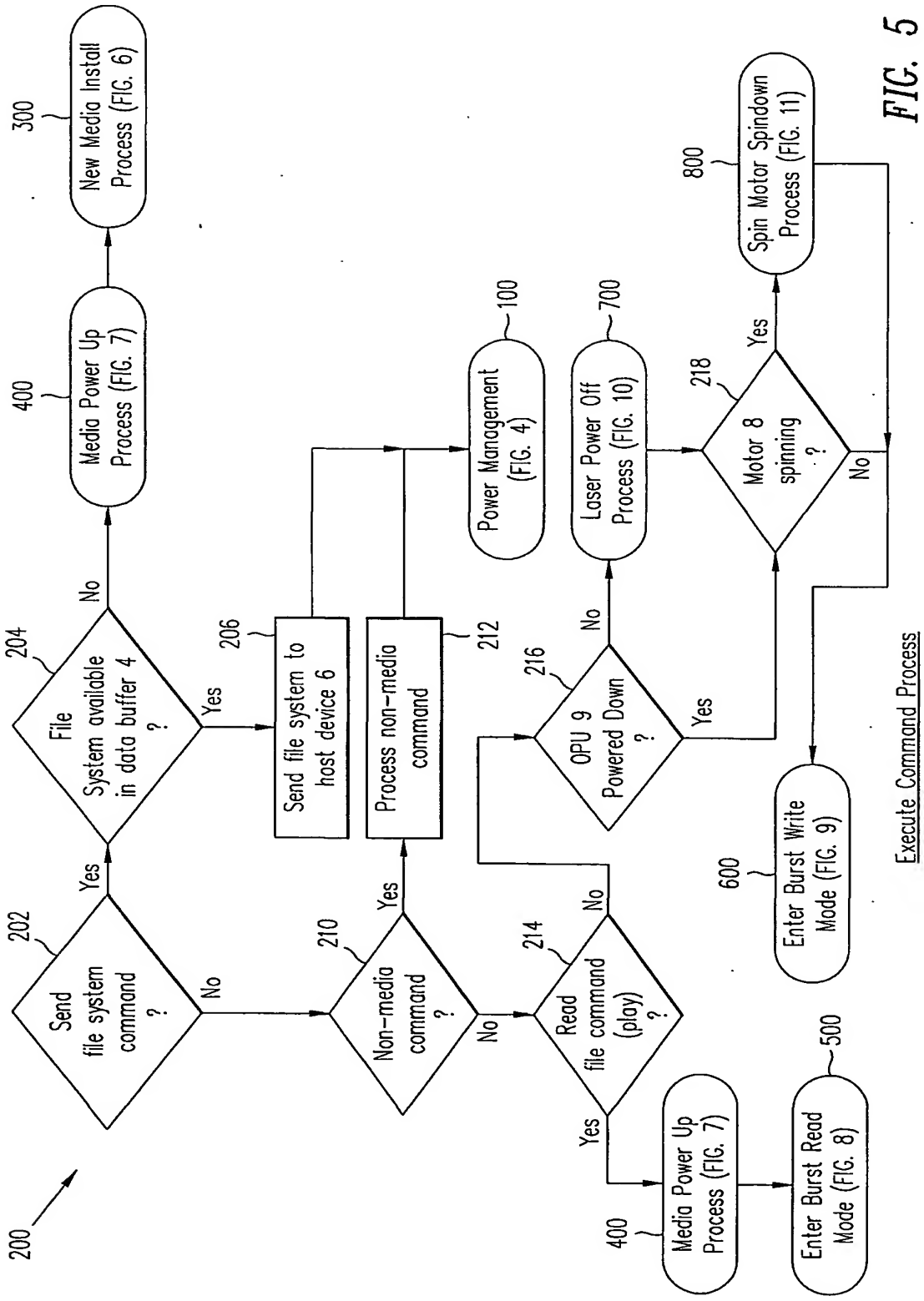
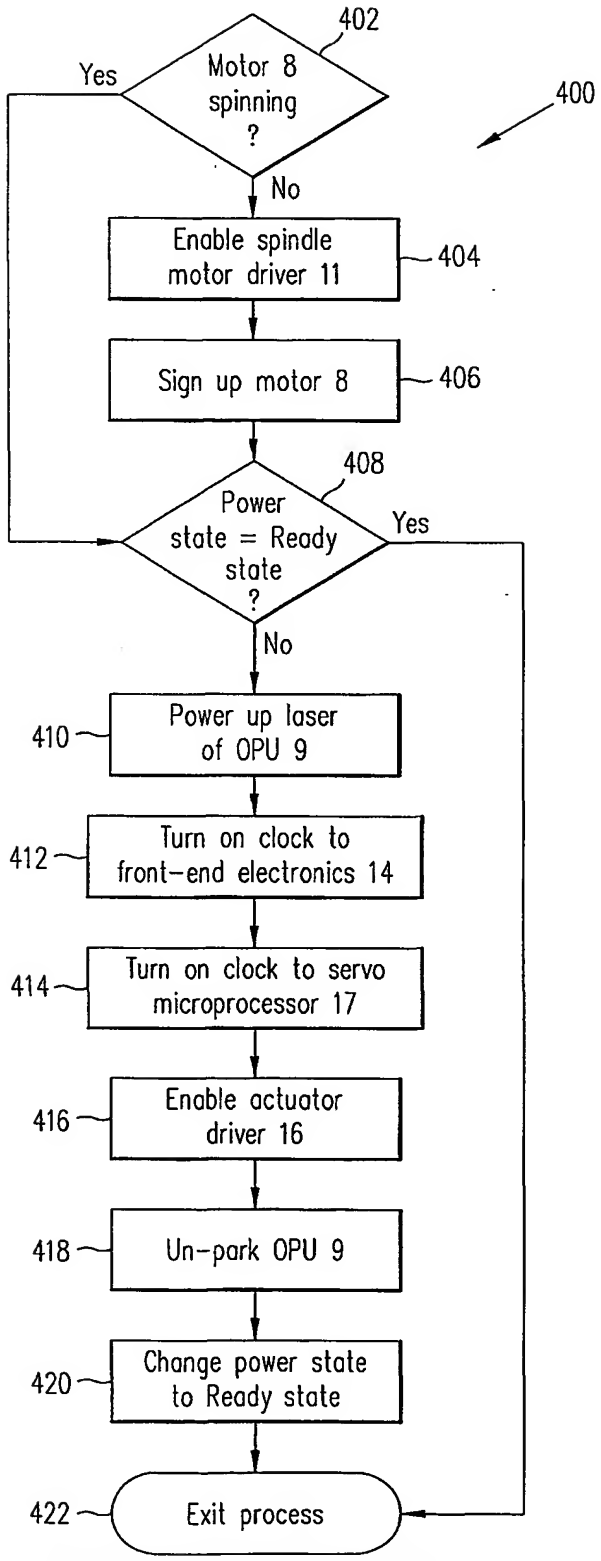
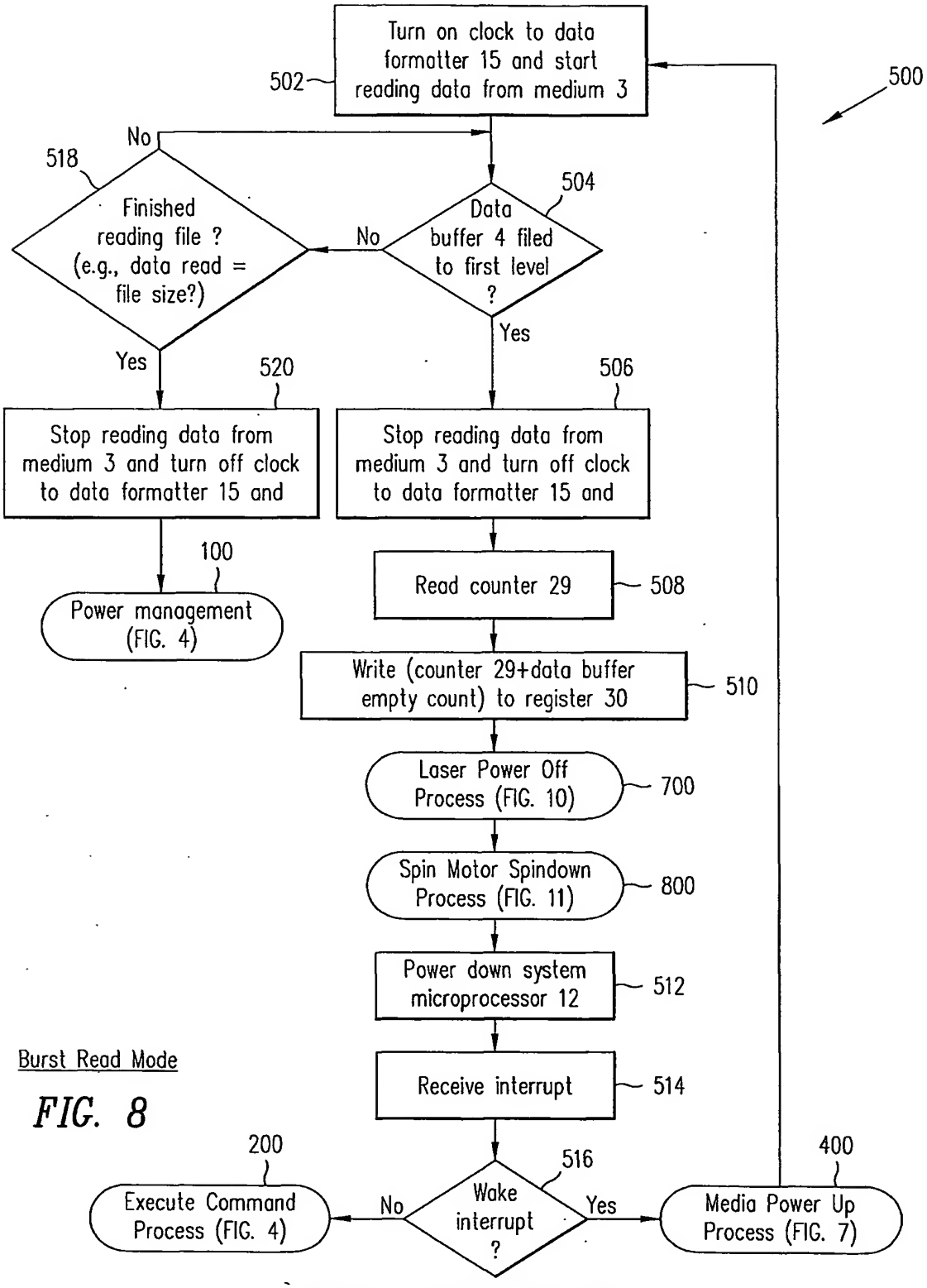

200
Send file system command ?
202
Yes
No
File System available in data buffer 4 ?
204
Yes
No
Media Power Up Process (FIG. 7)
400
New Media Install Process (FIG. 6)
300
Send file system to host device 6
206
Non-media command ?
210
Yes
No
Process non-media command
212
Power Management (FIG. 4)
100
Read file command (play) ?
214
Yes
No
OPU 9 Powered Down ?
216
Yes
No
Laser Power Off Process (FIG. 10)
700
Motor 8 spinning ?
218
Yes
No
Spin Motor Spindown Process (FIG. 11)
800
Media Power Up Process (FIG. 7)
400
Enter Burst Read Mode (FIG. 8)
500
Enter Burst Write Mode (FIG. 9)
600
Execute Command Process

FIG. 5




Media Power Up Process

*FIG. 7*




Burst Read Mode

*FIG. 8*

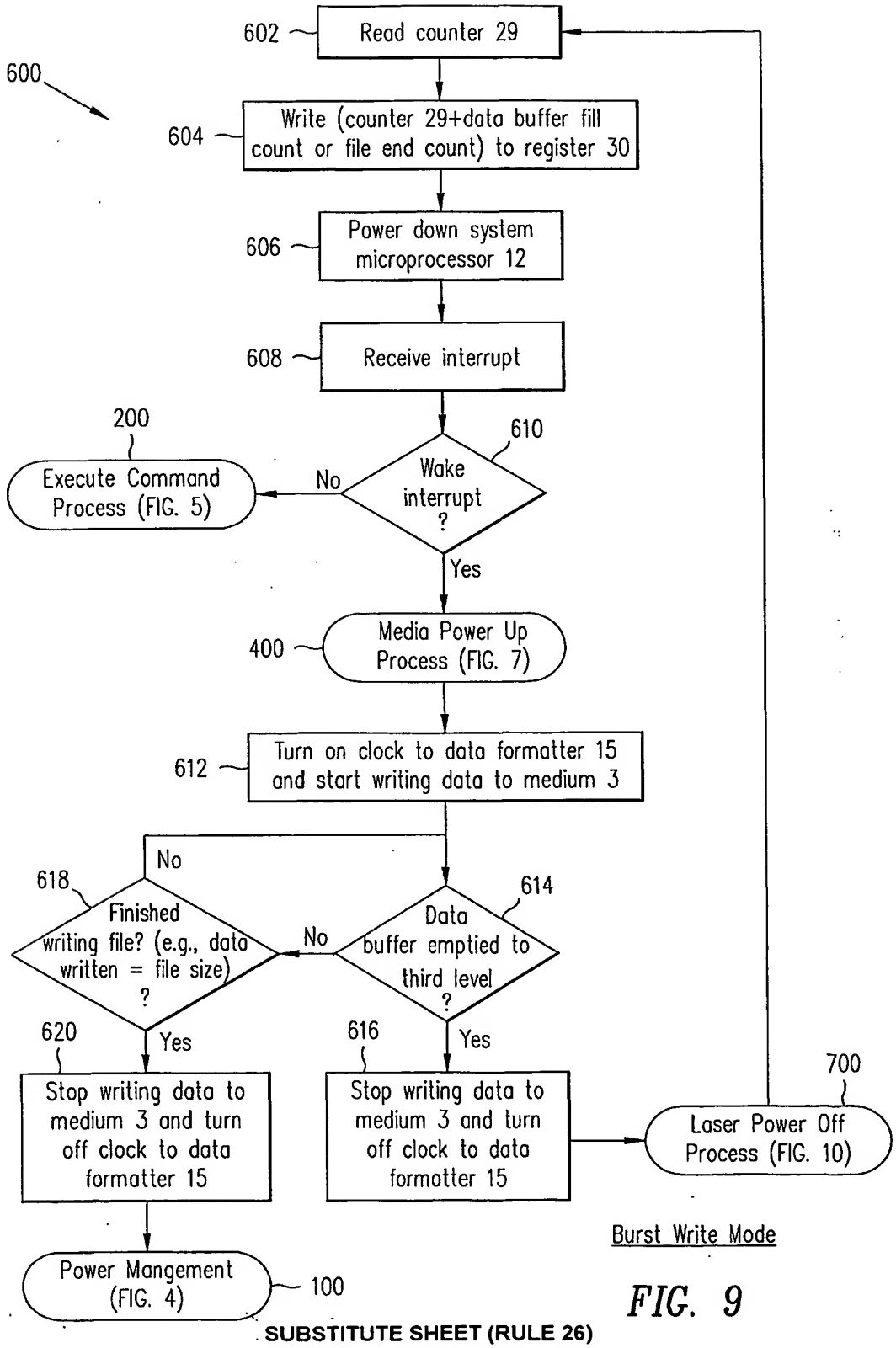
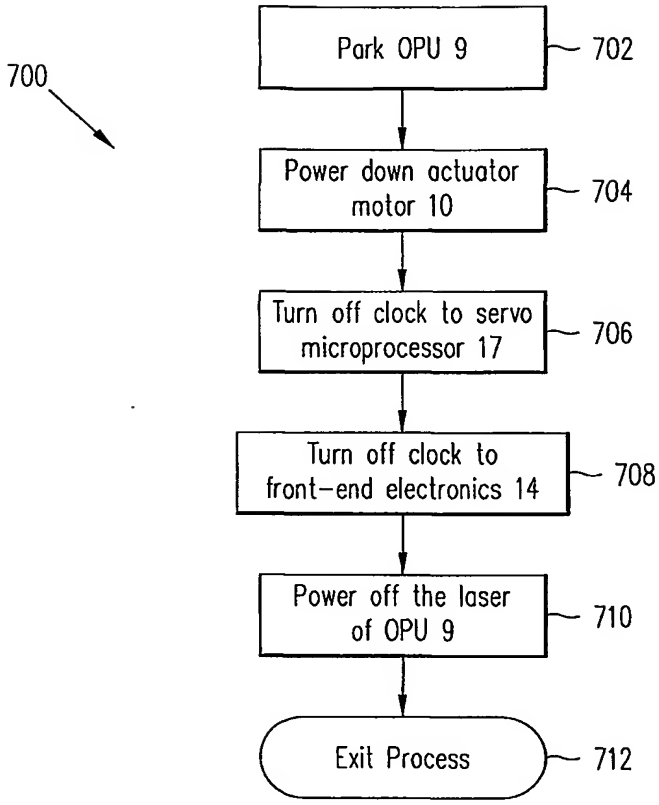
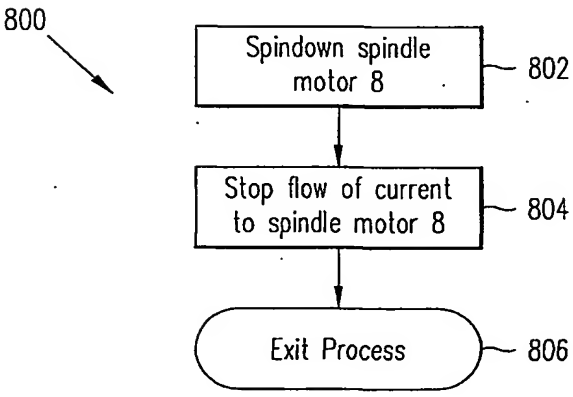

600
602
Read counter 29
604
Write (counter 29+data buffer fill count or file end count) to register 30
606
Power down system microprocessor 12
608
Receive interrupt
610
Wake interrupt ?
No
200
Execute Command Process (FIG. 5)
Yes
400
Media Power Up Process (FIG. 7)
612
Turn on clock to data formatter 15 and start writing data to medium 3
614
Data buffer emptied to third level ?
No
618
Finished writing file? (e.g., data written = file size) ?
No
Yes
620
Stop writing data to medium 3 and turn off clock to data formatter 15
Power Mangement (FIG. 4)
100
616
Yes
Stop writing data to medium 3 and turn off clock to data formatter 15
700
Laser Power Off Process (FIG. 10)
Burst Write Mode

FIG. 9




Laser Power Off Process

*FIG. 10*




Spin Motor Spindown Process

*FIG. 11*